CHAPTER 4
Once the selection of the technique of clustering is done, there are certain questions to be answered:

- What will be the optimal number of clusters?
- Which algorithm will be chosen as the best for customer segmentation?

The above addressed problems can be solved using validation. In statistics or data mining, a typical task is to learn a model from available data. Such a model may be a regression model or a classifier. The problem with evaluating such a model is that it may demonstrate adequate prediction capability on the training data, but might fail to predict future unseen data. Validation is a procedure for estimating the generalisation performance in this context. The idea for validation originated in the 1930s (Larson, 1931). In the paper one sample is used for regression and a second for prediction. Mosteller and Turkey (1963) and various other people further developed the idea. A clear statement of validation, which is similar to current version of k-fold cross-validation, first appeared in (Mosteller, & Turkey, 1968). In 1970s, both Stone (1974) and Geisser (1975) employed validation as means for choosing proper model parameters, as opposed to using validation purely for estimating model performance. Currently, validation is widely accepted in data mining and machine learning community, and serves as a standard procedure for performance estimation and model selection.

There are two possible goals in validation:

- To estimate performance of the learned model from available data using one algorithm. In other words, to gauge the generalisability of an algorithm.
• To compare the performance of two or more different algorithms and find out the best algorithm for the available data, or alternatively to compare the performance of two or more variants of a parameterised model.

The above two goals are highly related, since the second goal is automatically achieved if one knows the accurate estimates of performance. Given a sample of $N$ data instances and a learning algorithm $A$, the average validated accuracy of $A$ on these $N$ instances may be taken as an estimate for the accuracy of $A$ on unseen data when $A$ is trained on all $N$ instances. Alternatively if the end goal is to compare two learning algorithms, the performance samples obtained through cross-validation can be used to perform two-sample statistical hypothesis tests, comparing a pair of learning algorithms.

4.1 Different types of validation

Validating the performance of the models is the single most important step, if one can be chosen, in the process of data mining. One important mechanism for testing models is resampling. When selecting a validation technique, it is vital to keep in mind the purpose of such validation: to estimate the level of performance we may expect from models generated by our modeling process, when such models are run on future cases. There are different types of validation (Dwinnell, 2008):

i) **Apparent Performance**: The most obvious testing method is to simply execute the model on the very same data upon which it was built. The result is known as the *apparent* performance. The apparent performance is known to be statistically biased in an optimistic way. This is like giving out the answers to the test before administering the test.

At the extreme, a model could simply memorise the development observations and repeat them during testing. Assuming no mutually contradictory cases, such a system would deliver perfect validation performance.
The whole point in making a predictive model is so that said model may be used on future cases. What is desired is generalisation to new cases, not simple memorisation of historical ones.

Ultimately, there is no way to know precisely how optimistic apparent performance estimates are, rendering such performance measures largely useless.

Despite its hazards, calculation of the apparent performance is used as the final assessment of models with shocking frequency in industry. Many industries became one of its victims.

ii) **Holdout testing**: Given the dangers of apparent performance measures, one might logically reason that a model could be built using all presently available data, and get tested at some future point in time, after further observations had been collected. This idea makes perfect sense, but involves potentially considerable delay. Rather than wait for new data, holdout testing splits the data randomly into two sets: training (also called "in-sample") and testing (also called "out-of-sample"). This is the simplest form of resampling. Incidentally, it is not uncommon to stratify the assignment to training and testing groups, based on variables believed to be significant, including the dependent variables.

The idea here is, fit the model using the training data, and test it on the testing data.

Holdout testing provides an unbiased measure of performance, provided (and this caveat is rather important) that the test data is used only once to test the model. If the test data is used more than once to test the data, then all sets are off regarding the unbiased nature of the performance measure. In the event that another set of data is needed to make adjustments to the model (to
experiment with different numbers of predictors, for instance), a third randomly assigned data set, the validation set (also called the "tuning set") should be employed.

This simple test process works well in many instances in practice. Its biggest drawback is that it trades off training accuracy for testing accuracy. Typically, the data miner is faced with finite supply of data. Every observation which is moved to the testing set is no longer available for training.

In many cases, the primary interest is in evaluation of the model-generating process. Once it is what to expect from models that come from the process, one may apply one's modeling process to the entire data set (without regard to train/test designations) to construct the final model.

iii) **k-Fold Cross validation:** Smaller data sets force an uncomfortable choice on the modeler using holdout testing: either short-change model construction or short-change testing. One solution is to use k-fold cross-validation (sometimes referred to as simply “cross-validation”).

k-fold cross-validation builds on the idea of holdout testing in way by rotating data through the process. Data is again divided randomly into groups, but now k equal-sized groups are used. As with holdout testing, stratification is sometimes used to force the folds to be statistically similar. The train-test process is repeated k times, each time leaving a different segment of the data out, as the test set.

A common choice for k is 10, resulting in 10-fold cross-validation. In 10-fold cross-validation, the observations are randomly assigned to 10 groups. Ten separate models are built and tested on distinct data segments. The resulting 10 performance measures are unbiased since none of them was built with test data that was used during training. The single, final performance measurement
is taken as the mean of these 10 performance measures. The magic of this process is that during each fold, 90% of the data is available for training, yet the final performance metric is based on 100% of the data.

When \( k \) is equal to the number of observations, this process goes by the special name leave-one-out. While this may be tempting, there are good reasons for choosing \( k \) in the range of 5 to 10.

The good news with \( k \)-fold cross-validation is that reliable, unbiased testing may be performed on smaller data sets than would be possible with simple train-and-test holdout testing. The disadvantage of this method is that this process obviously requires much more computational effort than holdout testing.

As with holdout testing, once the modeling process has been evaluated, it may run over the entire data set to produce the final model.

### 4.2 Cluster Validation

Cluster validation refers to the technique whether a partition is correct and how to measure the correctness of a partition. Clustering algorithms are designed in such a way that it gives the best fit. However, the best fit sometimes may not be meaningful at all. The number of clusters might not be correct or the cluster shapes do not correspond to the actual groups in the data. Sometimes it might so happened that the data cannot be grouped in a meaningful way. One can distinguish two main approaches to determine the correct number of clusters in the data (Jansen, 2007):

i) We can define a large number of clusters. Successively reduce this number by combining clusters that have the same properties.

ii) Cluster the data for different values of \( c \) and validate the correctness of the obtained clusters with validation measures.
4.2.1 Validation Measures

This research makes use of the second approach. Some kinds of validity indices are usually adopted to measure the adequacy of a structure recovered through cluster analysis. In general, indices of cluster validity fall into one of the three categories:

- Some validity indices measure partition validity by evaluating the properties of the crisp structure imposed on the data by the clustering algorithm.
- In the case of fuzzy clustering algorithms, some validity indices such as partition coefficient and classification entropy use only the information of fuzzy membership grades to evaluate clustering results.
- The third category consists of validity indices that make use of not only the fuzzy membership grades but also the structure of the data.

To implement the comparison, we use the following validation measures (Balasko, Abonyi, and Balazs, 2006):

i) Partition Coefficient (PC): Defined by Bezdek, Partition Coefficient (PC) is defined as the sum of squares of values of the partition divided by the number of values. It is maximal if the partition is hard and reaches a minimum for \( u = 1/c \) when every object is equally assigned to every cluster. It is used to measure the amount by which one cluster overlaps the other.

\[
PC(c) = \frac{1}{N} \sum_{i=1}^{c} \sum_{j=1}^{N} (u_{ij})^2
\]

Where, \( u_{ij} \) is the membership of data point \( j \) in cluster \( i \).

Disadvantages of the partition coefficient are its monotonic decreasing with increasing \( n \) and the lack of direct connection to some property of the data themselves. To reduce this tendency we can define a normalised partition coefficient where the partition for uniform partitions are subtracted from the actual partition coefficient (Futschik and Kasabov, 2002)
ii) **Classification Entropy (CE):** It is a variation on the partition coefficient which calculates the membership values of the cluster. It is based on Shannon’s information theory.

\[
CE(c) = -\frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{N} u_{ij} \log(u_{ij})
\]  

(4.2)

If we have a crisp partition we have the largest information (i.e. the minimum entropy). Therefore, a partition with the minimum entropy is regarded as a good partition. Although this validity measure is based on Shannon’s information theory, it can be viewed as a measure for the fuzziness of cluster partition, which is very similar to the partition coefficients. Bezdek proves the relation \(0 \leq 1 - PC(c) \leq CE(c)\) for all probabilistic cluster partitions \(c\). The limitation of the classification entropy can be attributed to it appearing to be monotonic and to an extent, to the heuristic nature of the rationale underlying its formulation. The smallest \(CE(c)\) indicates a valid optimal partition.

iii) **Partition Index (PI):** It is the ratio of the sum of compactness and separation of the clusters. The individual cluster is measured with the cluster validation method which is Normalised by dividing it by the fuzzy cardinality of the cluster. The sum of the value for each individual cluster is used to receive the partition index.

\[
PI(c) = \frac{\sum_{c=1}^{C} \sum_{j=1}^{N} (u_{ij})^n \| x_j - v_i \|^2}{N \sum_{c=1}^{C} \sum_{j=1}^{N} \| v_j - v_i \|^2}
\]

(4.3)

PI is mainly used for the comparing of different partitions with the same number of clusters. A minor value of a PI means a better partitioning.

iv) **Separation Index (SI):** The separation index uses a minimum-distance separation to validate the partitioning.

\[
SI(c) = \frac{\sum_{i=1}^{C} \sum_{j=1}^{N} (u_{ij})^n \| x_j - v_i \|^2}{N \min_{i,k} \| v_i - v_k \|^2}
\]

(4.4)
where \( \| x_j - v_i \| \) denotes the Euclidean distance between the pattern \( x_j \) and the cluster centre \( v_i \). The separation measure \( S_I \) is based on fuzzy compactness and separation. While the numerator of Eq. 4.4 measures the total variance (or the compactness) of each fuzzy cluster, the denominator measures the separation defined as the minimum distance between cluster centres. The smallest \( S_I(c) \) indicates a valid optimal partition because \( u_{ij} \) will be high when \( \| x_j - v_i \| \) is low and well separated clusters will produce a high value of the minimum Euclidean distance between cluster centres.

\[ \text{v) Xie and Beni's Index (XB):} \] It is a method to quantify the proportion of the total deviation within the clusters and how the clusters are separated. The lowest value of the XB index should indicate the optimal number of clusters.

\[
XB(c) = \frac{\sum_{i=1}^{c} \sum_{j=1}^{N} (u_{ij})^2 \| x_j - v_i \|^2}{N \min_{i,k} \| x_j - v_i \|^2} \tag{4.5}
\]

\[ \text{vi) Dunn's Index (DI):} \] This index was originally designed for the identification of hard partitioning clustering. Therefore, the result of the clustering has to be recalculated.

\[
DI(c) = \min_{c \geq 2} \left\{ \min_{x \neq y} \left\{ \frac{\min_{x \neq y} d(x,y)}{\max_{x \neq y} d(x,y)} \right\} \right\} \tag{4.6}
\]

It is clear that if \( X \) contains compact and well-separated clusters, Dunn's index will be large, since the distance between the clusters is expected to be "large" and the diameter of the clusters is expected to be "small." Conversely, large values of Dunn's index indicate the presence of compact and well-separated clusters. The index \( DI(c) \) does not exhibit any trend with respect to \( c \), hence the maximum in the plot of \( DI(c) \) versus \( c \) can be used to indicate the number of clusters that underlie \( X \).

If \( DI(c) > 1 \) for a specific clustering, then this clustering contains compact and well-separated clusters. The main disadvantage of the Dunn's index is the very expensive computational complexity as \( c \) and \( N \) increase. Moreover, Dunn's
index is sensitive to the presence of noisy vectors in $X$, because these are likely to increase the value of the denominator of Eq. (4.6).

vii) Alternative Dunn's Index (ADI): To simplify the calculation of the Dunn index, the Alternative Dunn Index was designed. This will be the case when the dissimilarity between two clusters, measured with is rated in under bound by the triangle-inequality:

$$d(x, y) \geq |d(y, v_j) - d(x, v_j)|$$  \hspace{1cm} (4.7)

where $v_j$ represents the cluster center of the $j$-th cluster.

A point to be noted is that the Partition Coefficient and the Classification Entropy are only useful for fuzzy partitioned clustering. In case of fuzzy clusters the values of the Dunn’s Index and the Alternative Dunn Index are not reliable. This is caused by the repartitioning of the results with the hard partition method.